Remarks

These Remarks are in reply to the Office Action mailed March 26, 2003. Claims 1-30, 47-50, 83 and 84 were pending in the Application prior to the outstanding Office Action. In the Office Action, the Examiner rejected claims 1-30, 47-50, 83 and 84 under 35 U.S.C. §103(a) as allegedly being unpatentable over U.S. Patent No. 5,675,799 to Doktor in view of U.S. Patent No. 5,386,571 to Kurtz.

I. Claims 1 and 47

Claims 1 and 47 both claim the following features:

a) multiple operation records each storing data relating to one or more historical operation involving at least one entity, each said operation record comprising data recording the operation, and data defining a date associated with the operation, each said entity being an identifiable thing within a business or other undertaking to which information resulting from a transaction, measurement or other such assignment can be related; and

b) multiple entity records storing data indicating relationships between said entities, and each said relationship being associated with a historical period of validity.

It is defined in the claims that entity records store data indicating relationships

between entities. For example, a sales manager and an area of sales may be the two entities that have a relationship, i.e. the sales manager A is responsible for sales area C. The relationship is also associated with a historical period of validity, for example, the sales manager A was only responsible for sales area C during the year 2001. With this arrangement it is possible to update the business organisation stored in a database without having to go through the expensive and time consuming task of reformatting database structures.

On page 4 of the Office Action, it would seem that the Examiner concedes that Doktor does not explicitly disclose that each relationship is associated with a historical period of validity. The Examiner then asserts that Kurz implicitly discloses a period of validity. Applicants respectfully disagree with this assertion for the following reasons:

Within Kurz (col 5, lines 51 - 53) it is stated that: "The entity 'contract' may comprise the additional attributes date of signature and period of validity."

However, in Kurz, the single entity 'contract' has merely been assigned the attributes 'date of signature' and 'period of validity'. Each of these attributes will be a single data value. One of the attributes has by chance been given the arbitrary name of 'period of validity', which could, for example, alternatively (and more descriptively) have been called 'term of contract'. This attribute is merely used as a method of storing the actual term for which the legal terms of a contract are valid, such as, for example, 6 months, or 2 weeks, and is not

used for any other purpose. The Kurz document indicates that there are relationships between entities, as indicated, for example, by the interconnecting lines between the boxes shown in Figure 1. However, there is no implication or suggestion that the relationships between, for example, the entities 'contract' and 'description', and 'document' have been given a period of validity. The 'period of validity' associated with entity 'contract' is not applied to a relationship between the entity 'contract' and any of the other entities shown in the entity relational diagram. There is no mention in Kurz of storing data indicating an entity relationship between 'contract' and any other entities in the database. Kurz merely mentions that the attribute 'period of validity' has been assigned to the entity 'contract'. No other entities mentioned in Kurz are associated with the actual contract. In summary, there is no mention, or implication, of associating a historical period of validity with a relationship between the entities, with data indicating that relationship being stored in multiple entity records. Thus, a skilled person who happened to refer to Kurz would NOT take the inventive step of seeing the entity 'contract' as stored in this entity relationship diagram, stored with a one off attribute purely associated with that single entity, the attribute arbitrarily termed 'period of validity', and then arrive at the system as claimed when combining Kurz with Doktor. The skilled person would merely see an entity with an attribute assigned to it, and would not assume that a relationship between two entities has been given a period of validity.

In addition, the advantage of using a period of validity to store data before and after a change, and to store the change of the database structure over time (as described in the present application, page 6, line 23 - page 7, line 2), is not shown in Kurz.

In order to further distinguish between the present application and Kurz, an example is provided indicating how the invention of the present application could be used using the terms taken from Kurz (**however, Kurz does NOT provide such an example**). The present invention would provide multiple entity records that store data indicating relationships between the entities, each relationship being associated with a period of validity. The entities, taken from Kurz, could be the parties involved in the making of the contract. The relationship between the parties would be given a period of validity, for example, there may be a start period of validity indicating when the parties first started working together. Also, there may be an end time for the period of validity of the relationship between the parties when the parties no longer work together. This period of validity for the relationship between the parties (i.e. the entities) would be stored in the database. Changes to the periods of validity of the relationships are possible without the need to overhaul and restructure the whole database. For example, if one of the parties splits into two separate entities, the change in the structure of the organisation could be recorded by adding the new relationship between the two new entities, and amending the valid period of validity for the old relationships, for example, by amending the end time of the period of validity. Further, it would then be possible to use the period of validity of the relationship between the entities within the database in order to easily search through the database for the relevant periods in which requested data is stored. However, there is **no disclosure or implication** within Kurz that periods of validity for relationships between entities can be used in a database structure as indicated in this example, and so it is asserted that claims 1 and 47 are novel and non-obvious

in light of Kurz and Doktor.

As claims 1 and 47 are novel and non-obvious over Kurz and Doktor, it is asserted that all claims dependent on claims 1 and 47 (i.e., claims 2-30, and claims 48-50) are also patentable.

II. Claim 83

In addition to the arguments raised in the section above in relation to claims 1 and 47, which are also pertinent to claim 83, the following arguments should also be taken into account in relation to claim 83.

Claim 83 claims the following (emphasis added):

> A data processing system comprising a data storage device and a processor programmed to read data from, and write data to, said storage device, **in which said storage device stores a time variant data model to which data in a data structure conforms, the data model generated by the processor and representing the relationships between a plurality of classes of entities**, said storage device further storing:
>
> a) **multiple operation records each storing data relating to one or more historical operations involving at least one said entity conforming**

to one of said classes, each said operation record comprising data recording the operation, and data defining a date associated with the operation, each said entity being an identifiable thing within a business or other undertaking to which information resulting from a transaction, measurement or other such assignment can be related; and

b) **multiple entity records and association records which conform to the data model, each of the multiple entity records comprising an entity record for each said entity conforming to one of said classes**, said association records storing data indicating past or present relationships between a pair of said entities, and each said entity record containing data associating each said relationship with a historical period of validity.

This claims that a time variant data model and data structure are stored alongside each other within the storage device. The data structure is a particular example of the data model. The data model uses metadata to describe the classes of entities that make up the data model. Each of the entities stored in the data structure conform to one of the classes of entities within the data model. Figure 7 of the present application shows an example of the data model, and Figures 8a and 8b show examples of the data structure.

Neither Doktor nor Kurtz indicate that a data model and a data structure can be stored alongside each other, and so do not have the advantage of being able to modify the data model and the data structure in order to keep, for example, a business organisation database

up to date with minimum cost and effort.

It is therefore asserted that claim 83 is novel and non-obvious over Kurz and Doktor.

III. Claims 48 and 84

With regards to the Examiner's comment in Section 2 of the Office Action it is stated that the "Applicant's arguments with respect to claims 1-30, 47-50 and 83-84 have been considered but are moot in view of the new ground(s) of rejection." However, because Kurz was not used in any rejection of claims 48 and 84, Applicants assert that their previous arguments should still stand (i.e., are not moot)

Claim 48 claims as follows (emphasis added):

> A data processing system comprising a data storage device and a processor programmed to read data from, and write data to, said storage device, in which said storage device stores multiple operation records each storing data relating to one or more historical operation involving at least one entity, each said entity being an identifiable thing within a business or other undertaking to which information resulting from a transaction, measurement or other such assignment can be related; and multiple entity records storing

data indicating relationships between said entities, **wherein the entity records comprise a hierarchical structure, in which at least a first entity record relates to a specific entity, and a second to a more generic entity encompassing said specific entity, said entity records including link data linking said first and second entity records whereby to allow said processor to traverse said hierarchy**, said processor being arranged to generate output data by inputting instructions defining one or more selected entity dimensions across which said output data is to be distributed.

The discussion in the introduction of Doktor is merely describing, at a fundamental level (column 3, line 46), how data is stored in a computing device, i.e. that data is stored as '1's or '0's, with an address tag attached to the data to show where the data is stored. Alternatively, the data may be represented by an address, indicating where the data is stored.

Doktor then states:

"Some bit strings may function as address pointers, rather than as the final pieces of 'real' information which a database user wishes to obtain. The address pointers are used to create so-called 'threaded list' organizations of data wherein logical links between a first informational 'object' (first piece of real data) and a second informational 'object' (second piece of real data) are established by a chain of direct or indirect pointers."

In other words, there are strings of bits that merely point to a physical or memory location where the data is stored. The strings of bits are not the 'real data' required, but are an address indicating where the real data is stored. The threaded list allows a user to be directed from one piece of data to another piece of data in a sequential manner. The two pieces of real data are not formed in a hierarchical structure wherein one piece of real data is a generic piece of data and a second piece of real data is a more specific piece of real data, as claimed in claim 48.

For example, Doktor goes on to say at column 5, lines 14 – 21 that:

> "A large variety of different relations can therefore be established between a first piece of real data (e.g., a first person's name) and a second piece of real data (e.g., a second person's name) simply by changing the sort keys that are stored in the separate sort column (e.g., who is older than whom, who is taller, etc.)."

Therefore, from this description, the two pieces of real data mentioned at column 3, line 65 – 66 may each be a person's name, such as for example "Mr. Harry W. Jones" and "Mrs. Barbara R. Smith", as shown at column 5, lines 7 – 9. These two pieces of real data can in no way be considered to be a specific entity, and a generic entity encompassing the specific entity, as claimed in claim 48, but are rather two specific entities, neither

encompassing the other.

Further, in Doktor at column 4, line 11 it is stated that:

> "A serial sequence of rows ... is then defined by linking one row to another according to a predefined sorting algorithm using threaded list techniques."

This is a very time consuming method of looking for data, and in no way describes a hierarchical structure. Further, there is a requirement for a predefined sorting algorithm in order to sort through the data. The use of this sorting algorithm is described in Doktor in column 4, lines 31 – 47. A table is sorted in order according to a sorting algorithm, which in turn allows the system to search down the sort column to find the relevant piece of data. For example, a column may be sorted in alphabetical or numerical order. The system described in Doktor does not allow a processor to traverse entity records that form a hierarchical structure, as in the present invention, and so claim 48 is novel and non-obvious over Doktor.

Further, as claims 49 and 50 are dependent on claim 48, they are also novel and non-obvious for the same reasons given above.

The arguments raised in relation to claim 48 are also pertinent to claim 84. In addition to the arguments above, the following arguments should also be taken into account.

Claim 84 claims as follows (emphasis added):

> A data processing system comprising a data storage device and a processor programmed to read data from, and write data to, said storage device, in which said storage device stores multiple operation records each storing data relating to one or more historical operation involving at least one entity, each said entity being an identifiable thing within a business or other undertaking to which information resulting from a transaction, measurement or other such assignment can be related; and multiple entity records storing data indicating relationships between said entities, wherein the entity records comprise a hierarchical structure, in which at least a first entity record relates to a specific entity, and a second to a more generic entity encompassing said specific entity, said entity records including link data linking said first and second entity records whereby to allow said processor to traverse said hierarchy, said processor being arranged to generate output data by inputting instructions defining one or more selected entity dimensions across which said output data is to be distributed; and **if all required said operation records do not relate to entities of the dimension to which the operation records relate, the processor is programmed to determine, from said entity records, a hierarchically higher level entity dimension and to repeat said determination and, in the event that all required said operation records relate to said hierarchically higher level, to use said**

hierarchically higher entity instead of said selected entity in selecting said subset of operation records.

Page 36, line 3 to page 38, line 2 of the current application gives an example of how the system traverses the hierarchical levels in order to obtain as much useful data as possible when answering a query. This allows a user to obtain data from an organisation over a period of time, whether the organisation has undergone fundamental restructuring or not. Any restructuring could, in standard relational databases, result in the loss of the relevant data required. Whereas in the system of the present invention, the data is stored in a manner such that data relevant before and after the organisational restructure is kept, and the hierarchical structure can be traversed by the system in order to extract relevant data from before and after any organisational restructuring.

In Doktor (column 5, lines 37 – 42) it is stated that:

"Relational database tables are normally organized to create implied set and subset 'relations' between their respective items of pre-stored information. The lowest level subsets are stored in base tables and higher level sets are built by defining, in other tables, combinations of keys which point to the base tables."

The arrangement described in Doktor (col. 4, line 58 – col. 6 line 52) provides a base table with raw data, and separate relational tables that have pointers that point to the raw data.

An access control program determines which relational tables to look at, determines the relevant 'real' data by looking at the correct 'key' in the base table, and then determines a relationship between the different pieces of 'real' data by the sequence in which the tables are accessed.

Therefore, in Doktor, the base table is merely a look up table holding raw data. This allows a data value to be changed in the base table so that every time a pointer from a relational table points to that data, it will read the updated data. This is so that changes to the data only need to carried out once in the base table and not in several tables that use that data value.

The system described in Doktor is not the same, nor implies the same meaning, as a processor running through a list of entity records in a hierarchical structure in order to determine a list of requested operational records, and then subsequently changing the hierarchical level at which to search through the entity records if the requested operational records could not be found at the first hierarchical level, as claimed in claim 84. Doktor describes a base table that is used to merely store raw data and is not used to search for operational records. Therefore, claim 84 is novel and non-obvious over Doktor.

Finally, the applicants have reviewed Kurz, and believe that Kurz does not teach the deficiencies of Doktor in relation to claims 48 and 84.

Conclusion

In light of the above, it is respectfully submitted that all of the claims now pending in the subject patent application should be allowable. Reconsideration and allowance of all claims is, therefore, respectfully requested. The Examiner is respectfully requested to telephone the undersigned if he can assist in any way in expediting issuance of a patent.

The Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 06-1325 for any matter in connection with this response, including any fee for extension of time, which may be required.

Respectfully submitted,

Date: June 24, 2003

By: _____
Jeffrey R. Kurin
Reg. No. 41,132

Fliesler Dubb Meyer & Lovejoy LLP
Four Embarcadero Center, Fourth Floor
San Francisco, California 94111-4156
Telephone: (415) 362-3800